Tasking Sensor Networks

Johannes Gehrke

Cornell University

Research Associate: Manuel Calimlim

PhD Students: Rohit Ananthakrishna, Adina Costea, Abhinandan Das, Alexandre Evfimievski, Manpreet Singh, Yong Yao

Background

Characteristics of future battlespace environments and homeland defense monitoring systems:

- Thousands or millions of small-scale sensor nodes
- Nodes combine multiple sensing and computation capabilities
- Limited resources at the sensors: Network, power, CPU

Application requirements:

- Scalability
- Complex monitoring tasks, multiple user types, multiple missions, multiple systems
- Survivability under stress and under attack
- High-confidence in measured events and predictions
- Easy deployment and zero-overhead administration

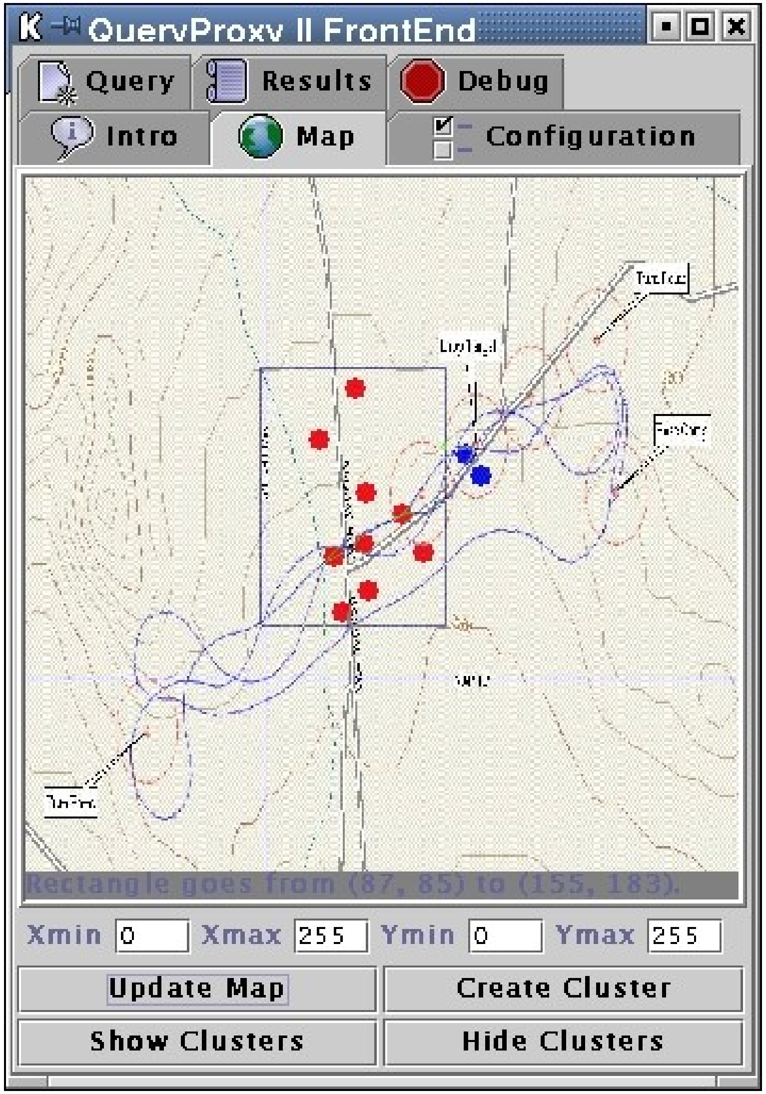
Flexible Decision Support

Traditional

Procedural addressing of individual sensor nodes; user specifies how task executes, data is processed centrally.

SensIT

Complex declarative querying and tasking. User isolated from "how the network works", in-network distributed processing.



Querying: Model

Time	Value
12	82
13	83

Time	Value
13	82
15	83

Time	Value
13	82
15	84

Time	Value
14	79
15	83

Time	Value
13	82
15	83

Time	Value
13	80
16	83

Example Queries

- Snapshot queries:
  - What is the concentration of chemical X in the northeast quadrant?

```
SELECT AVG(R.sensor.concentration)
FROM Relation R
WHERE R.sensor.loc in (50,50,100,100)
```

  - In which area is the concentration of chemical X higher than the average concentration?

```
SELECT AVG(R.sensor.concentration)
FROM Relation R
GROUP BY R.area
HAVING AVG(R.sensor.concentration) >
  (SELECT AVG(R.sensor.concentration)
   FROM Relation R
   GROUP BY R.area)
```

Example Queries (Contd.)

- Long-running queries
  - Notify me over the next hour whenever the concentration of chemical X in an area is higher than my security threshold.
    ```
    SELECT R.sensor.area, AVG(R.sensor.concentration)
    FROM Relation R
    WHERE R.sensor.loc in rectangle
    GROUP BY R.sensor.area
    DURATION (now,now+3600)
    ```
  - Notify me if a TEL is driving south on Route 13
  - Notify me if a TEL and a T72 cross
- Archival queries
  - Periodic data collection for offline analysis

Goals

- Declarative, high-level tasking
- User is shielded from network characteristics
  - Changes in network conditions
  - Changes in power availability
  - Node movement
- System optimizes resources
  - High-level optimization of multiple queries
  - Trade accuracy versus resource usage versus timeliness of query answer

Technical Challenges

- Scale of the system
- Constraints
  - Power
  - Communication
  - Computation
- Constant change
- Distribution and decentralization
- Uncertainty from sensor measurements

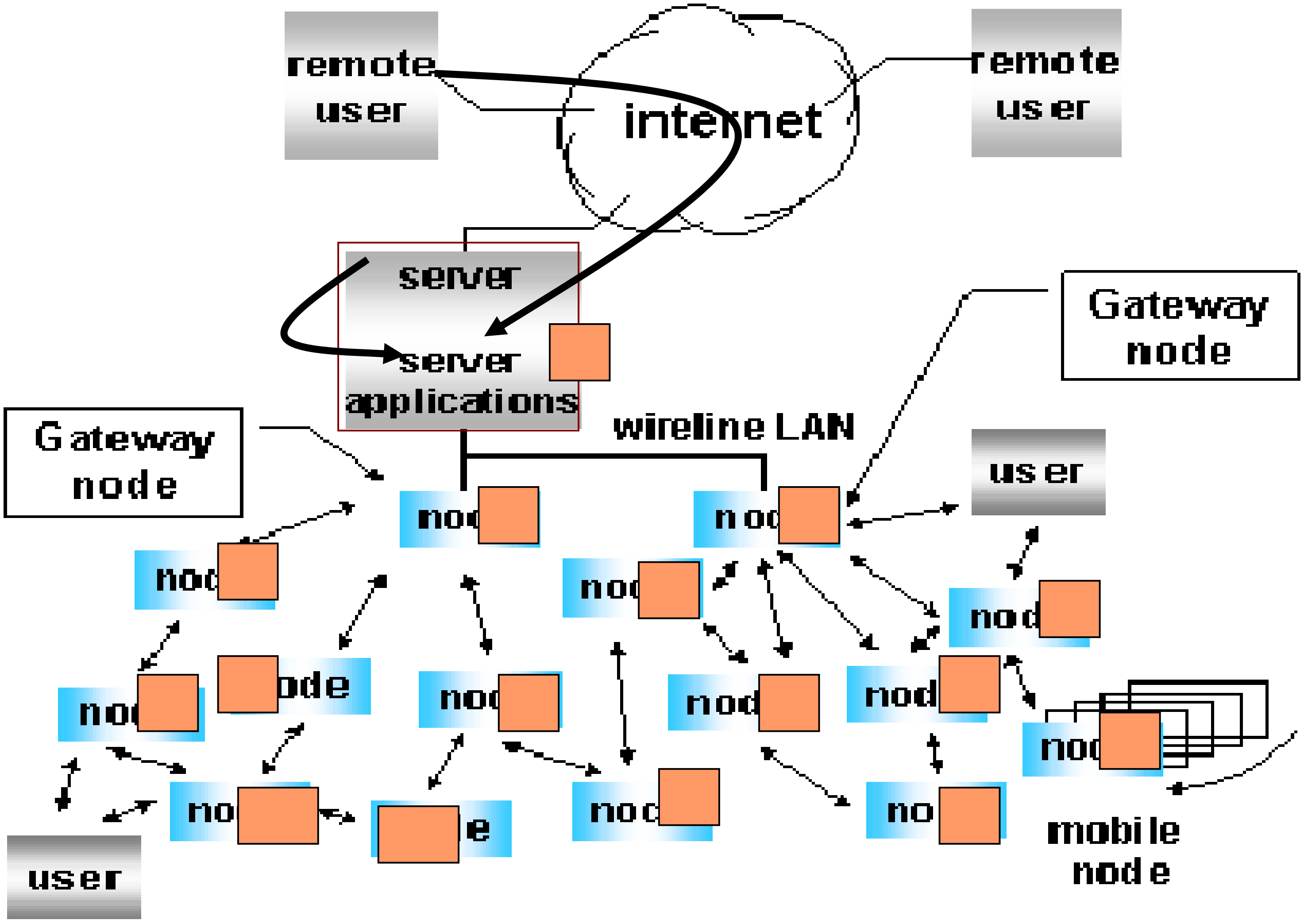
Cornell Contributions

- Scalable query processing architecture
- High-level complex tasking (queries!)
  - Declarative XQuery-related high-level query language; can be generated directly from GUI
  - All-XML interfaces and communication structures
- Sensor query processing
  - In-network query processing
  - Data stream processing
  - New probabilistic data model
  - Fault-tolerant adaptive query processing

Talk Outline

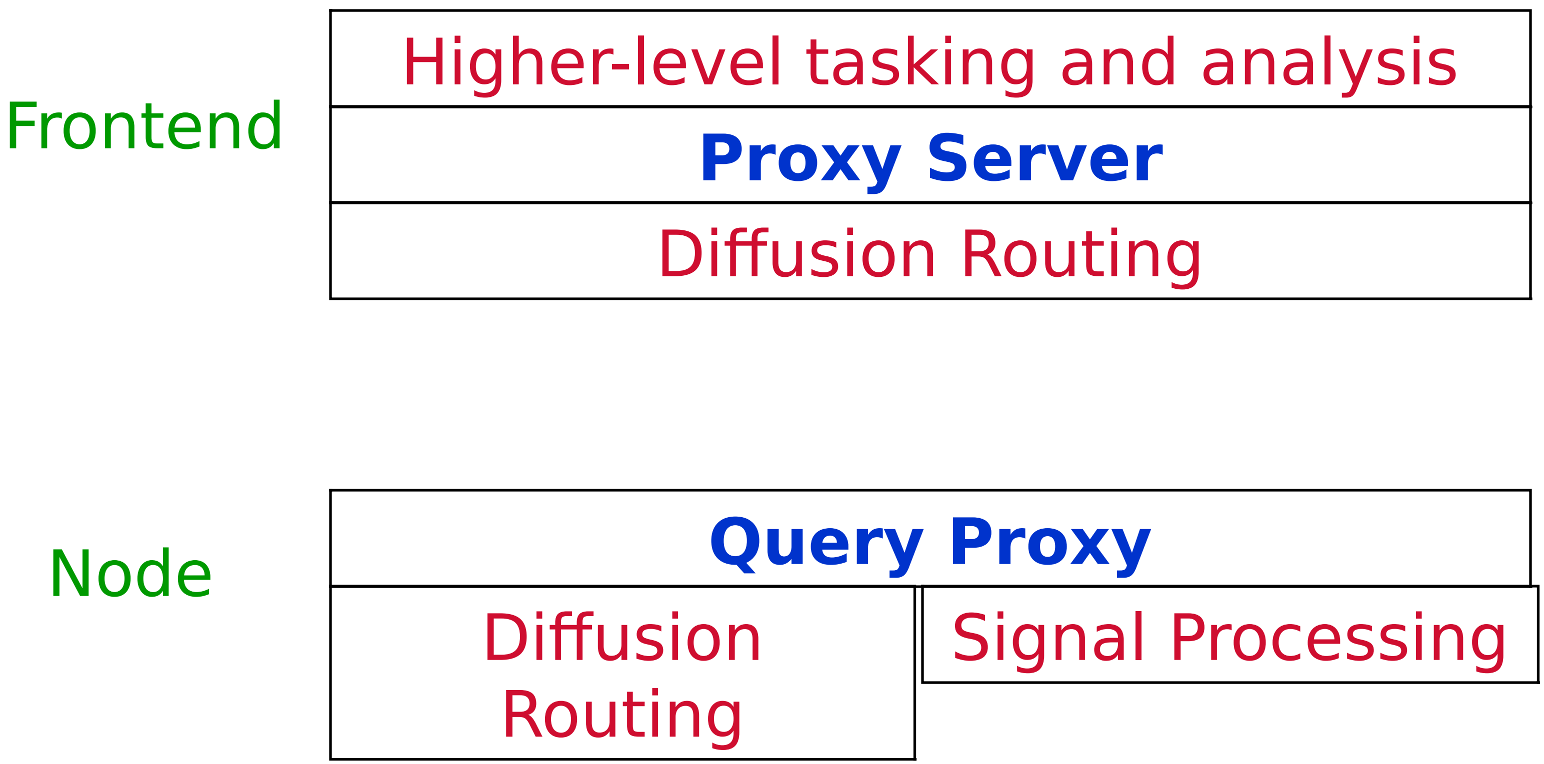
- Querying sensor networks
- Technical discussion
  - Scalable query processing architectures
  - High-level tasking
  - Sensor query processing
- Outlook
- Conclusions

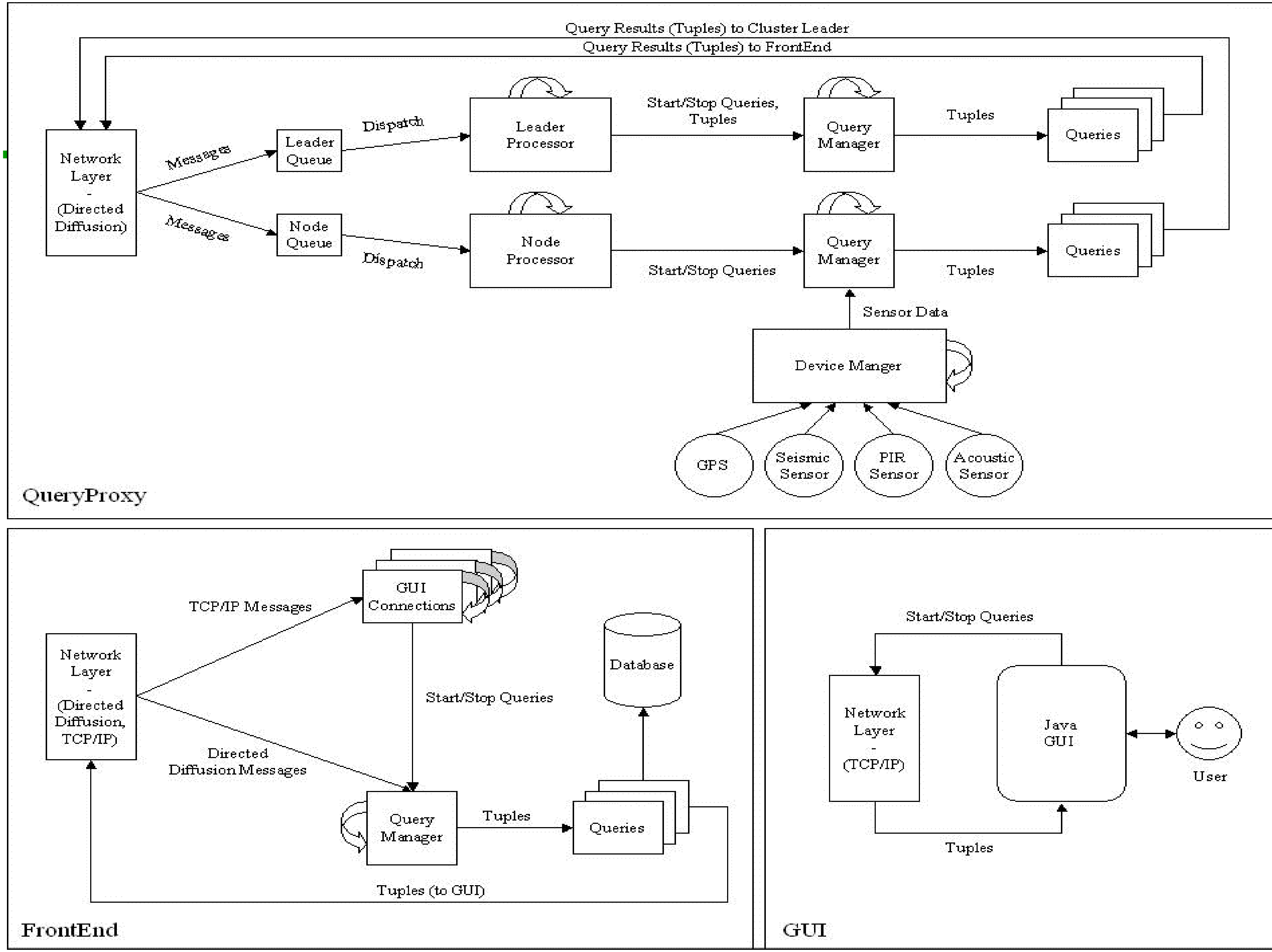
The Cornell Cougar System:



The Cornell Cougar System

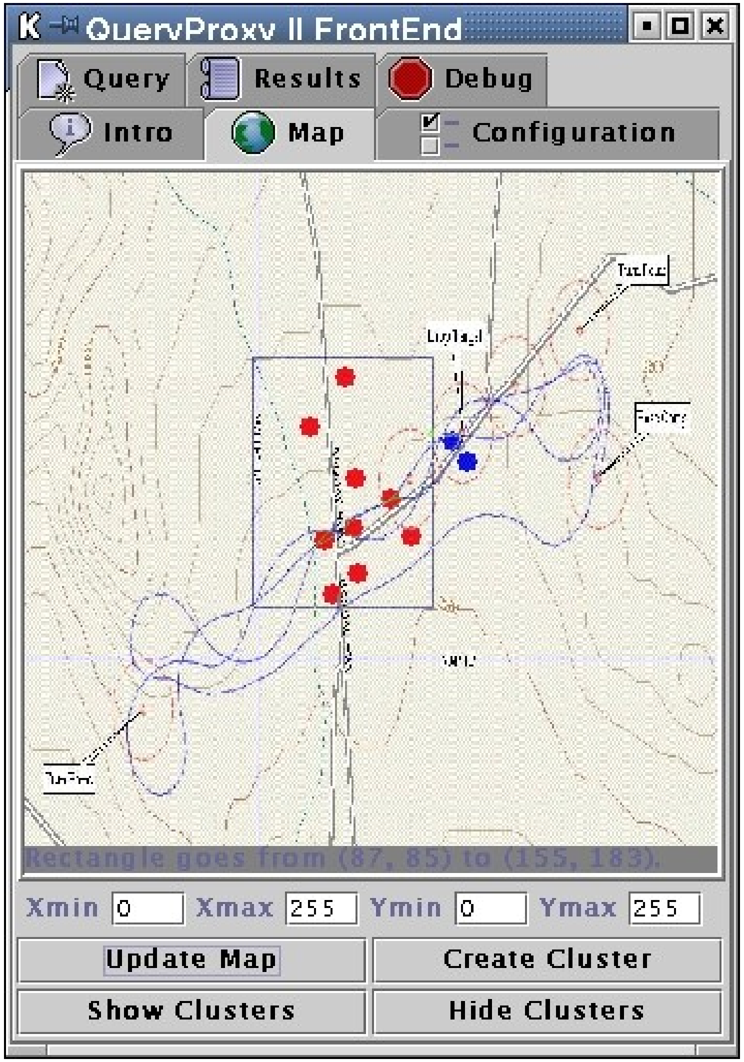
Cougar in the (simplified) SensIT Architecture





Query Results (Tuples) to Cluster Leader
Query Results (Tuples) to FrontEnd
Network Layer - (Directed Diffusion)
Messages
Messages
Leader Queue
Node Queue
Dispatch
Dispatch
Leader Processor
Node Processor
Start/Stop Queries, Tuples
Start/Stop Queries
Query Manager
Query Manager
Tuples
Tuples
Queries
Queries
Sensor Data
Device Manger
GPS
Seismic Sensor
PIR Sensor
Acoustic Sensor
QueryProxy
Network Layer - (Directed Diffusion, TCP/IP)
TCP/IP Messages
GUI Connections
Start/Stop Queries
Database
Directed Diffusion Messages
Query Manager
Tuples
Queries
Tuples (to GUI)
FrontEnd
Start/Stop Queries
Network Layer - (TCP/IP)
Java GUI
User
Tuples
GUI

Sample User Interface



High-Level Complex Tasking

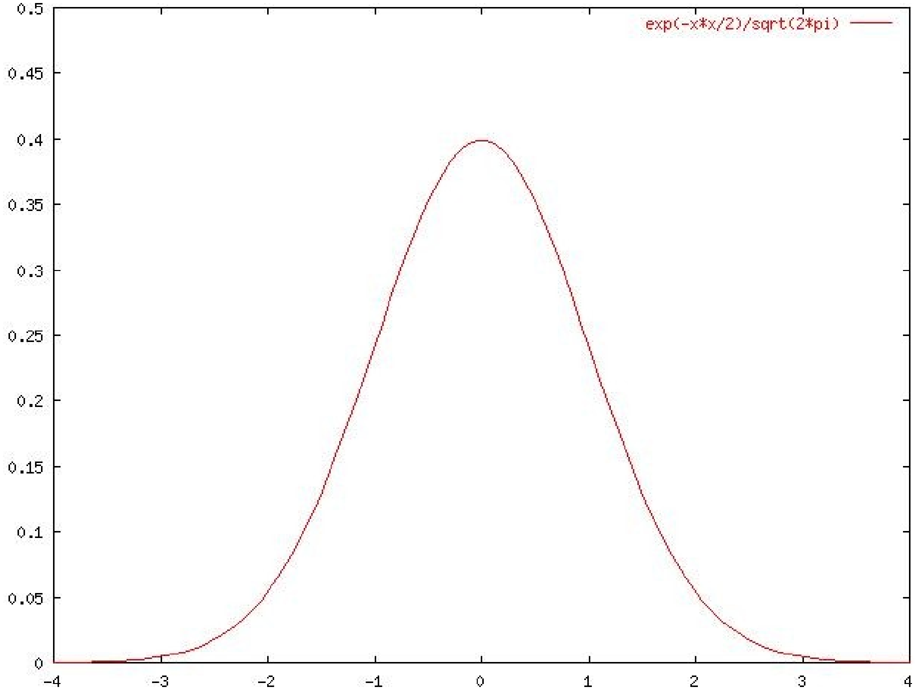
- Query language based on XQuery allows complex declarative tasking
  - User is shielded from physical network properties
  - GUI generates declarative queries
  - System optimizes queries, re-optimizes queries, adapts to physical network conditions

Sensor Query Processing

- Data model
- In-network processing
- Records arrive in high-speed data streams
- Environmental conditions are constantly changing

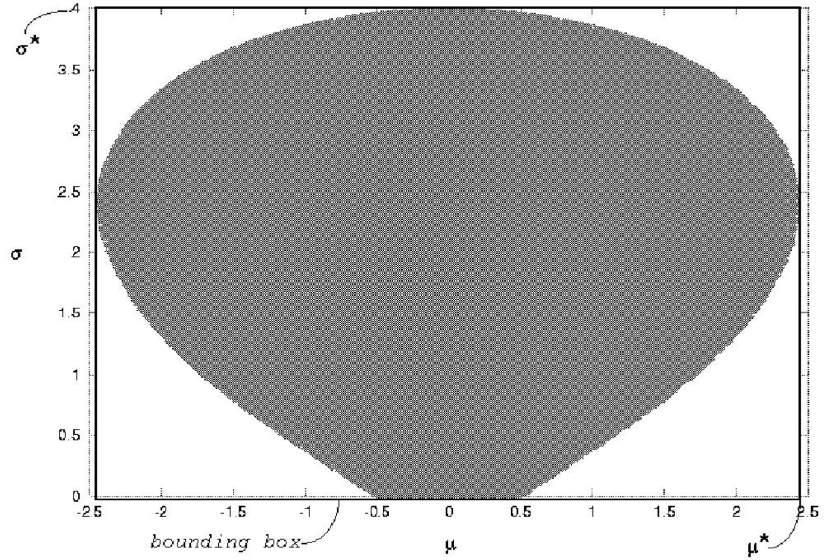
GADT: Relational Data for Sensors

- We extended relational DBMSs with a new Gaussian data type. Gaussians are now first-class values.



Evaluating GADT Queries

- GADT instances that satisfy a query can be simply visualized as a subset of the 2D plane
- Example: R.a. Prob([-0.5,0.5])>0.1
- We can use database indexing techniques t process such queries



GADT Data Type

Implementation level: GADT Operators

- Selection
- Projection
- Join

Conceptual level: Theory

- Measure-theoretic formulation of probabilistic data
- New framework for probabilistic data

Sensor Query Processing

- Data model
- In-network processing
- Records arrive in high-speed data streams
- Environmental conditions are constantly changing

In-Network Processing

What is distributed in-network processing?

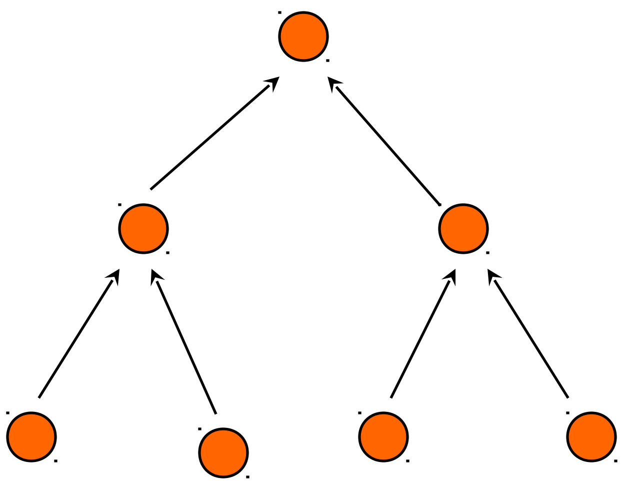
- Processing at the nodes where the data originates (the “source nodes”)
- Processing at “intermediate nodes”
- Processing only at relevant nodes

Why is this hard?

- Scale
- Constantly changing conditions
- Meta-data management
- Fault tolerance

Processing at Intermediate Nodes (1)

- Onto which nodes should we place query processing operators?

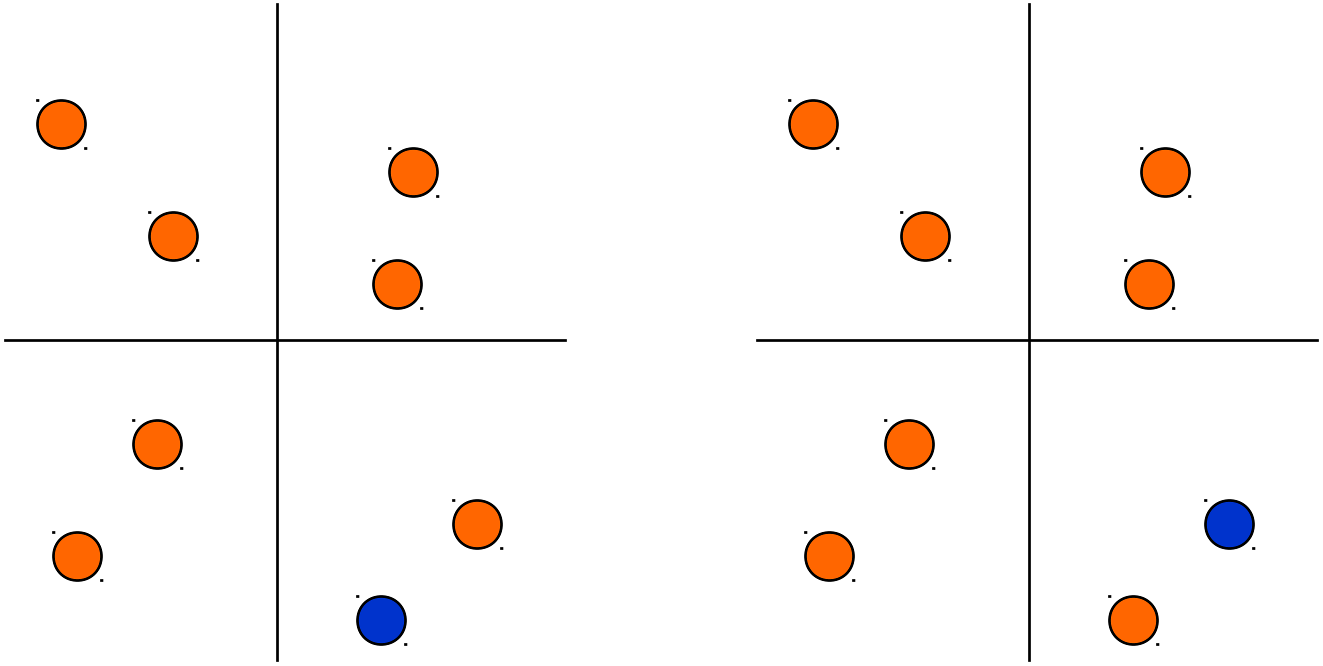


Processing at Intermediate Nodes (2)

- Several new aggregation algorithms that make use of processing at the intermediate nodes
  - Simple spanning tree aggregation
  - Fault-tolerant super-node spanning tree aggregation
- Simulation results and results from working implementation:
  - Reduces network traffic
  - Increases battery life of the nodes
  - Scales gracefully with number of queries and number of nodes

Distributed Processing

- Switch intermediate processing based on available power.



In-Network Data Stream Processing

- Examples:
  - Quantiles with limited memory
    What was the median concentration of chemical X in this area over the last five minutes?
  - Correlated aggregates with limited memory
    During the last five minutes, where was the concentration of chemical X in this area higher than the average?

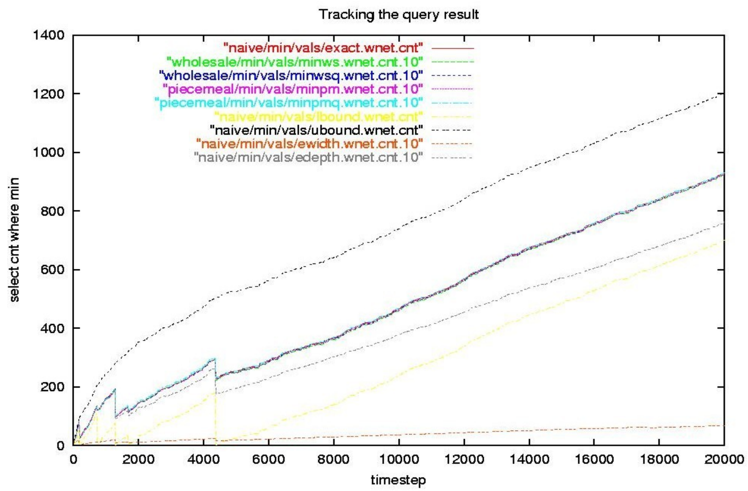
In-Network Data Stream Processing

- Why are aggregates with limited memory hard?

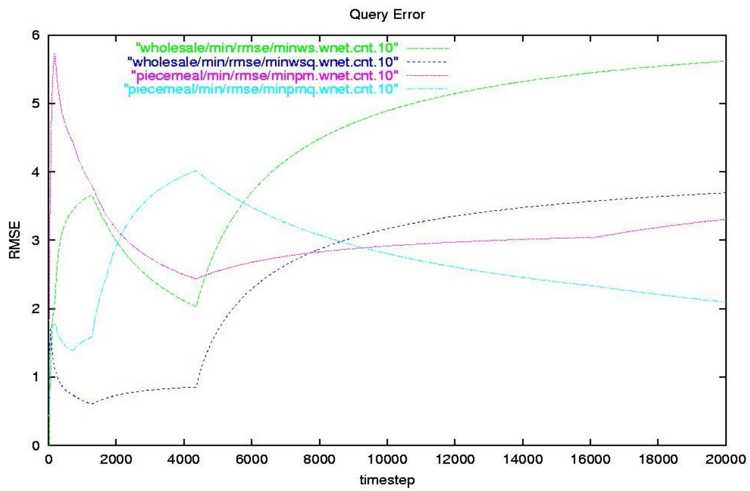
Our solution:

- Hierarchical, distributed algorithm, provable approximation guarantees, limited amount of memory.

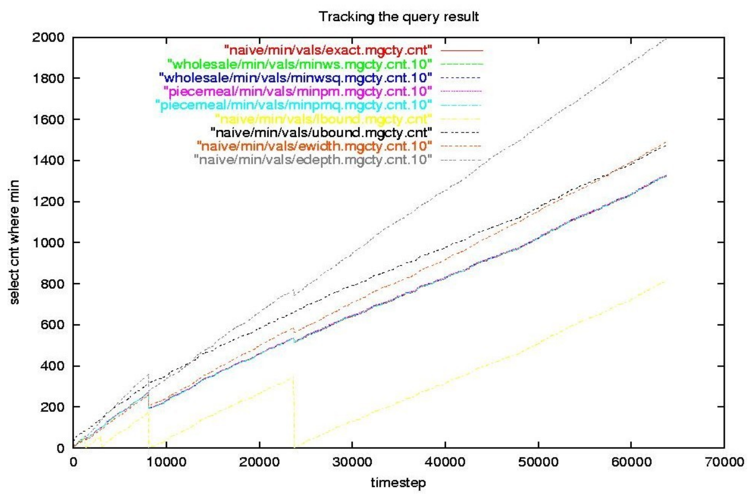
In-Network Processing



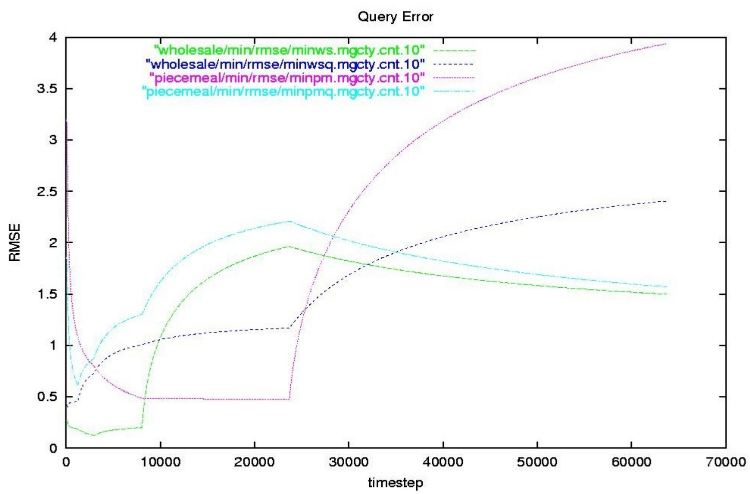
(a) tracking value for USAGE



(b) $RMSE_i$ for USAGE



(c) tracking value for MGCTY



(d) $RMSE_i$ for MGCTY

Example 1: Distributed Data Streams

Simple example: How many detections match?

Location	Type
(10,12)	A
(13,15)	B
(11,13)	C

Type	Speed
D	80
B	50
A	60

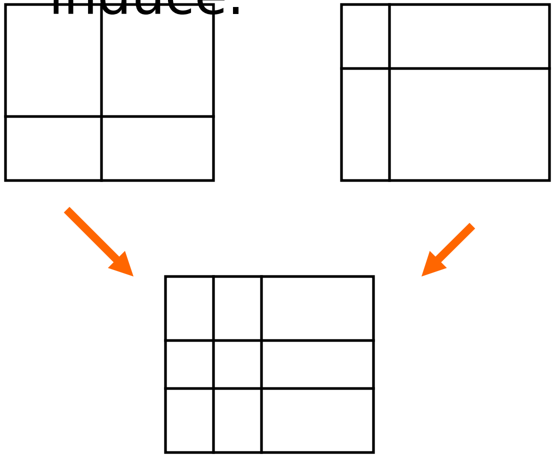
Techniques:

- k-wise independent random variables
- Histograms
- Other statistical techniques

Example 2: Change Detection

Approach 1:

Define difference metric ("deviation") at the data mining model level. Compare datasets through difference in the data mining models they induce.



Approach 2:

Mine hidden concepts from data streams. Monitor change of concepts.

0 0 1 0…
1 1 0 1…
0 1 1 1…

0 1 0 1…
1 0 0 0…
0 1 1 1…
1 0 0 1…
1 0 1 0…
0 1 1 0…
1 0 1 1…
0 0 1 1…
0 1 1 1…
1 1 0 1…

Talk Outline

- Querying sensor networks
- Technical discussion
  - Scalable query processing architectures
  - High-level tasking
  - Sensor query processing
- Outlook
- Conclusions

Where Do We Stand?

November 2000:

- Demonstrated basic query processing in November 2000 experiments
- Integrated with ISI diffusion routing
- Motivated major component of filters in diffusion API for in-network processing
- Demonstration at Intel Continuum Computing Conference

November 2001/January 2002:

- Developmental demo of query processing system at 29 Palms in November 2001 (integrated with ISI diffusion)
- Experimental demo for January 2002 PI meeting: Integrated with ISI diffusion routing, BAE Systems, Fantastic Data, ISI-West
- Integration work for AFRL

Publications Since Last PI Meeting

- V. Ganti, J. Gehrke, R. Ramakrishnan, and W.-Y. Loh. A Framework for Change Detection. Journal of Computer and Systems Science, 2001.
- J. Gehrke, F. Korn, and D. Srivastava. On Computing Correlated Aggregates Over Continual Data Streams. 2001 ACM SIGMOD Conference.
- Z. Chen, J. Gehrke, and F. Korn. Query optimization in compressed database systems. 2001 ACM SIGMOD Conference.
- T. Faradjian, J. Gehrke, and P. Bonnet. GADT: A Probability Space ADT For Representing and Querying the Physical World. 2002 IEEE ICDE Conference.

Under submission:

- Computing Complex Aggregates over Data Streams
- Which Aggregates Cannot be Approximated Well Over Data Streams?
- Adaptive Query Processing in Heterogeneous Environments
- A Framework for Physical Database Design
- Least Expected Cost Query Optimization

Impact

What will be the impact on national security and DoD?

- Continuous intelligence gathering at several orders larger magnitude
- Fast event notification
- High-level programming interface (“queries”)
- Establish a system infrastructure for sensor networks community

Integrate query processing into system infrastructure (embedded monitoring)

Plans for Remainder of Contract

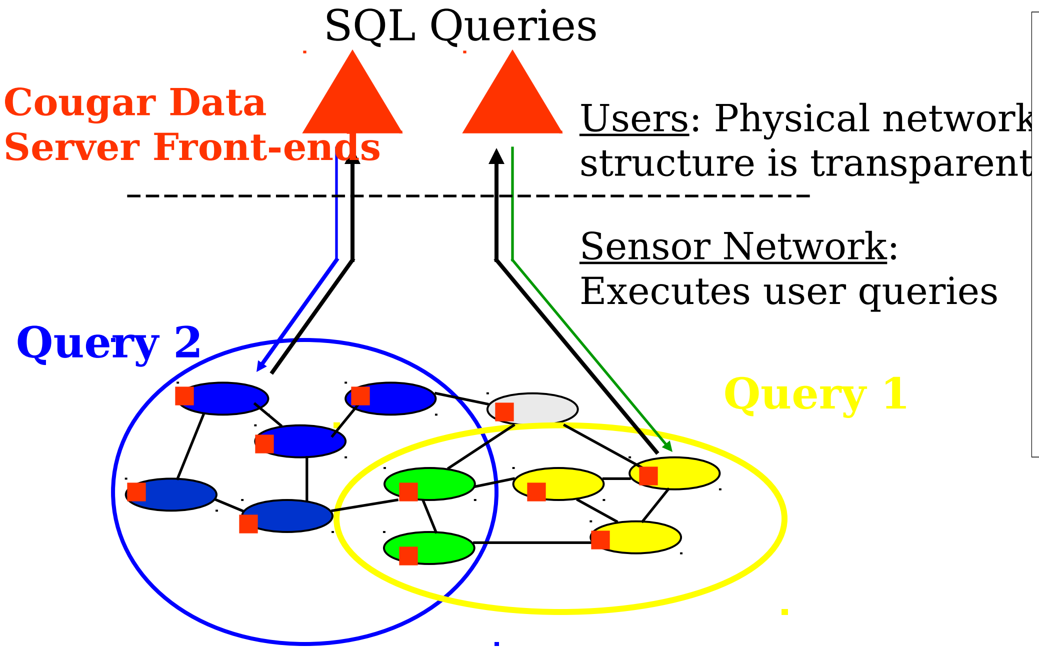
- Participation in large-scale experimental demo
- Demonstrate:
  - Reduced network traffic and reduced energy usage through in-network processing
  - Scalability with number of nodes
  - Scalability with number of queries

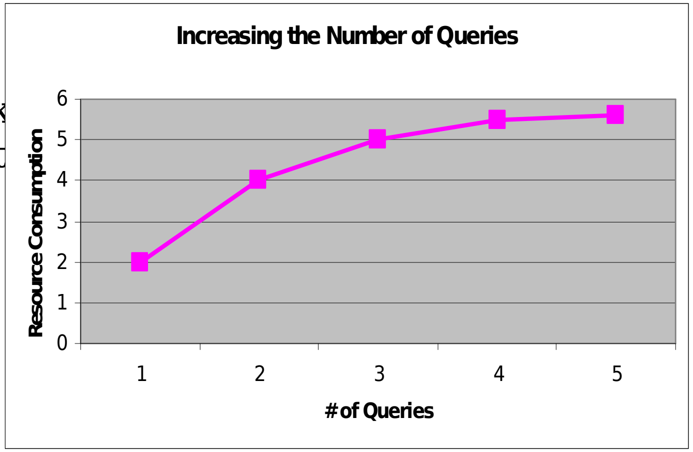
Outlook

- Multi-query optimization
- Triggers
- Historical and predictive queries
- Information assurance
- Internetworking for homeland defense

Multi-Query Optimization

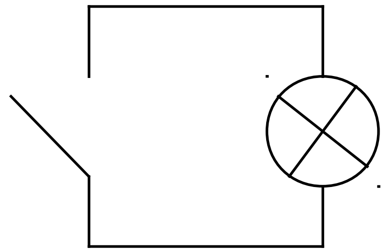
- Scenario: Multiple related, but slightly different queries
- Goal: Save power and communication
- Challenge: Combining multiple queries, finding common query parts





In-Network Geo-Spatial Triggers

- Database concepts:
  - Condition ("I am tracking a T-80")
  - Event ("It enters the northeast battle zone")
  - Action ("Turn on the cameras and alert commander")
- Why is this hard?
  - Current database systems choke at tens of triggers
  - Here we will have >100,000 personal triggers
- Technical Challenges:
  - **In-network** trigger management
  - Consistency of triggers
  - Scalability



Predictive Queries

- How many vehicles went by between 0600 and 0800?
- When is the vehicle going to reach the intersection?

- Technical challenges:
  - On-node distributed query processing and storage
  - Efficient compression of past events
  - Memory management and background archival
  - Prediction models right at the nodes

Information Assurance

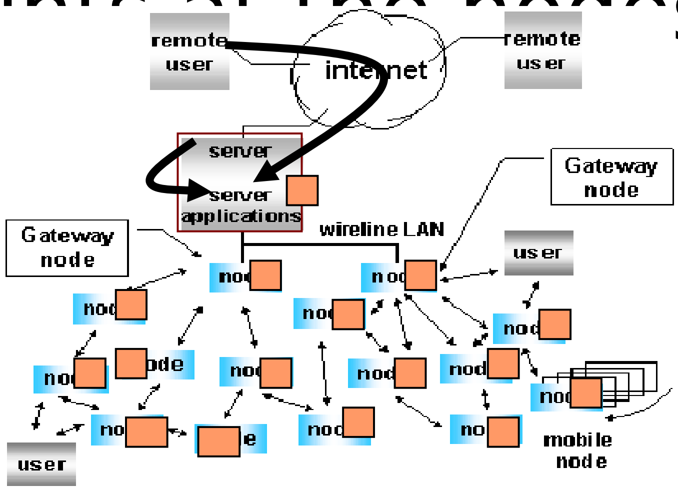
- Sensor failure
  - How do we know about broken sensors?
  - How to we compensate for broken sensors?
  - Can we predict sensor replacement needs?
  - Zero administration
  - 24/7/365 system uptime
- Sensor placement
  - Where should we place sensors?
  - Redundancy versus accuracy versus resource usage

Internetworking for Homeland Defense

- Integrate the physical world with other intelligence
  - The loop goes both ways → Open architectures, standard data and knowledge exchange (XML-based)
- Technical challenges:
  - Seamless fixed/mobile device interaction
  - Data integration
  - Scalability – both number of nodes and amount of data collected
  - Knowledge discovery and data mining

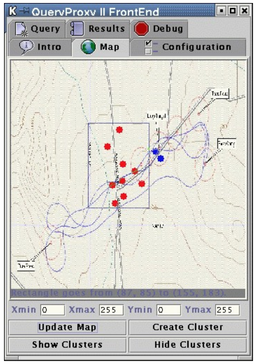
Summary

Distributed, highly scalable, fault-tolerant, energy-efficient query processing techniques that scale to large number of nodes and queries and works under tight resource constraints at the nodes.



Questions?

http://www.cs.cornell.edu/database/



The Cougar Team: Manuel Calimlim (Research Associate), Rohit Ananthakrishna, Zhiyuan Chen, Abhinandan Das, Alexandre Evfimievski, Yong Yao (PhD students)